eval manual page - Built-In Commands

tcl.tk/man/tcl/TclCmd/eval.htm

NAME

eval — Evaluate a Tcl script

SYNOPSIS

eval *arg* ?*arg ...*?

DESCRIPTION

Eval takes one or more arguments, which together comprise a Tcl script containing one or more commands. **Eval** concatenates all its arguments in the same fashion as the **concat** command, passes the concatenated string to the Tcl interpreter recursively, and returns the result of that evaluation (or any error generated by it). Note that the **list** command quotes sequences of words in such a way that they are not further expanded by the **eval** command.

EXAMPLES

Often, it is useful to store a fragment of a script in a variable and execute it later on with extra values appended. This technique is used in a number of places throughout the Tcl core (e.g. in **fcopy**, **lsort** and **trace** command callbacks). This example shows how to do this using core Tcl commands:

```
set script {
    puts "logging now"
    lappend $myCurrentLogVar
}
set myCurrentLogVar log1
# Set up a switch of logging variable part way through!
after 20000 set myCurrentLogVar log2

for {set i 0} {$i<10} {incr i} {
    # Introduce a random delay
    after [expr {int(5000 * rand())}]
    update    ;# Check for the asynch log switch
    eval $script $i [clock clicks]
}
```

Note that in the most common case (where the script fragment is actually just a list of words forming a command prefix), it is better to use **{*}$script** when doing this sort of invocation pattern. It is less general than the **eval** command, and hence easier to make robust in practice. The following procedure acts in a way that is analogous to the **lappend** command, except it inserts the argument values at the start of the list in the variable:

```
proc lprepend {varName args} {
    upvar 1 $varName var
    # Ensure that the variable exists and contains a list
    lappend var
    # Now we insert all the arguments in one go
    set var [eval [list linsert $var 0] $args]
}
```

However, the last line would now normally be written without **eval**, like this:

```
set var [linsert $var 0 {*}$args]
```